$2.25
September, 1983
Vol. 3 NO. 9

M.A.C.E. JOURNAL

"People For Computers"

* PM GRAPHICS II
* MODIFY YOUR 810
* ACTION REVIEW
* VISICALC CHART
* STAR MERCHANTS
* ACROSS THE POND
* BEST OF BAKER STREET BYTES

ABC's of PM Graphics Part II

Several people asked questions about part I. Here are the questions and the answers: q1."Where are figures 1-3?" The missing figures appear at the bottom of this article. (Figures 2 and 3 have been combined.) q2."What is the advantage of using strings?" One of the strongest points of ATARI Basic is strings. The string technique will perform multiple pokes at machine language speed! q3."How can I learn about all those memory locations?" The easiest way is to buy a memory map. Check at your local computer store.

This month we will see how the players can be moved and also how to put missiles on the screen. The lines in the listing must be added to the lines from part I. THESE LINES WILL NOT RUN WITHOUT PART I.

Lines 1210-1220 assign values to an array that correspond to the vertical position of a falling object. The ball is set in motion in line 1250. The ball is first erased (P0$=PM$), the horizontal position is set (POKE HP0,X), then the ball is drawn in the new vertical position (P0$[Y[C]]=BALL$). Of course, the human eye/brain is too slow to see the ball erased. In fact, without the delay (F.W=1 TO 10:N.W) the ball would move much too quickly! When the SELECT key is pushed the value of "FLAG" is 1 and the sideways position of the ball "X" is increased. This causes the ball to move from left to right. Push the OPTION key to exit the loop.

Lines 1330-1350 show how vertical motion can be achieved using a paddle. Line 1340 assures that the loop will be exited when the SELECT key is pushed.

Horizontal motion of a player is even easier. Line 1440 shows how easy it is to move the blue bird with a paddle. Notice how fast the bird can move and that it can be positioned beyond the visible screen.

Motion under joystick control is the next demo. Since reading the stick is more complicated, this routine will be slower. To compensate for this, the joystick read and move is placed near the beginning of the program. LOOPS WILL RUN FASTER NEAR THE BEGINNING OF A PROGRAM.

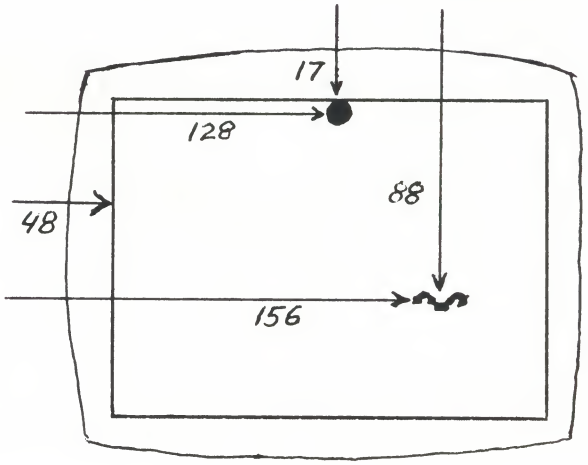
The horizontal position of the man is "X" and the vertical position is "Y". The sideways position will be changed in increments of 4 and the vertical position will be changed by 3. Moving the man up or down by 3 would not erase the old position from the player string P1$. If you move upwards his legs would stretch! Moving him down would cause his head to stretch! The solution to this problem is simple.

One solution would be to erase the man before plotting him in the new position. I have used a slightly different solution. Define a new string (M$) that contains the old shape (MAN$) and also contains 3 blank lines above and below the shape. Now when the vertical position is changed, the blank lines will remove the unwanted picture. Lines 30 and 40 also insure that the X and Y values will not go too far to cause an error. Line 50 provides a means of exiting the loop.

Lines 2000-2100 show how to build and position the missiles. Lines 2260 and 2270 show how to move the missiles vertically. Notice that the string B$ is the shape of the ball (BALL$) with a blank line before and after the ball shape. (See line 2250.) This concludes part II.

Again, please send me your questions and comments to Mark Davids, 21825 O'Connor, St. Clair Shores, MI. 48080

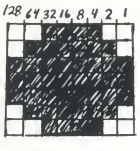
POSITIONING THE PLAYERS



```
20 S=STICK(0):P1$(Y)=M$:POKE HP1,X
30 NX=X+4*((S>4 AND S<8)*(X<250)-(S>8
AND S<12)*(X>5))
40 NY=Y+3*((S=5 OR S=9 OR
S=13)*(Y<108)-(S=6 OR S=10 OR S=14)*(Y>5))
50 X=NX:Y=NY:IF PEEK(53279)<>5 THEN 20
60 GOTO 2000:REM * SELECT PUSHED
1200 GOSUB WAIT
1210 DIM Y(13):RESTORE 1220:FOR C=1 TO
13:READ Z:Y(C)=Z:NEXT C
1220 DATA 17,20,27,40,57,80,
107,107,80,57,40,27,20
1230 X=50:POKE HP0,X:? :? "  Push START
to start ball bouncing.":GOSUB WAIT
1240 ? :? "  Push SELECT for sideways
motion.":? :LIST 1250,1280
1245 ? :? " Push OPTION to continue."
1250 FOR C=1 TO 13:P0$=PM$:POKE
HP0,X:P0$(Y(C))=BALL$:FOR W=1 TO
10:NEXT W
1260 IF PEEK(53279)=5 THEN FLAG=1
1265 IF PEEK(53279)=3 THEN 1290
1270 IF FLAG=1 THEN X=X+5:IF X>240
THEN X=10
1280 NEXT C:GOTO 1250
1290 ? CHR$(125):P0$=PM$:REM * ERASE
BALL
1300 GOSUB WAIT
1310 ? :? "  Use paddle for vertical motion."
1320 ? :? "  PUSH SELECT TO
CONTINUE.":LIST 1330,1350
1330 Y=100-INT(PADDLE(0)/2.5):P1$(1,
Y-1)=PM$:P1$(Y)=MAN$:P1$(Y+11)=PM$
1340 IF PEEK(53279)=5 THEN 1390
1350 Y2=100-INT(PADDLE(0)/2.5):GOTO
1330+10*(Y2=Y)
1390 P3$=PM$:REM * ERASE BUILDING
1400 ? CHR$(125):? :? " PADDLE controls
horizontal position.":? :? :LIST 1440
1410 P3$(50)=BIRD$:POKE COL3,7*16+10:
POKE WP3,1:REM * A BLUE BIRD
1430 POSITION 8,20:? "Push START to
continue."
1440 X=20+PADDLE(0):POKE HP3,X:IF
PEEK(53279)<>6 THEN 1440
1450 POKE HP3,50
1480 DIM M$(19):M$=PM$:M$(4)=MAN$:
M$(16)=PM$:P1$=PM$:X=120:Y=60
1500 REM * JOYSTICK MOVES MAN
1510 ? CHR$(125):? :? "  USE JOYSTICK TO
MOVE MAN.":LIST 20,60
1520 ? :? "  Push SELECT for next part."
1530 GOTO 20:REM * JOYSTICK MOVE
ROUTINE
2000 ? CHR$(125):? :? " All missiles will
now have the same  colour and live in the
same string  called MIS$"
2010 ? :? "  Here is how to put a shape into
the  string and set the colour:":LIST 2030
2030 MIS$(100)=BALL$:COLMIS=711:POKE
COLMIS,12*16+10:REM * THE COLOUR
ADDRESS
2040 ? "Each missile has its own horizontal
address and must be separately poked":? "
Watch closely !!"
2050 LIST 2060,2100
2060 HMIS0=53252:HMIS1=53253:
HMIS2=53254:HMIS3=53255
2070 POKE HMIS0,170:FOR W=1 TO
200:NEXT W
2080 POKE HMIS1,168:FOR W=1 TO
200:NEXT W
2090 POKE HMIS2,166:FOR W=1 TO
200:NEXT W
2100 POKE HMIS3,164:FOR W=1 TO
200:NEXT W
2190 ? :? "Push START to continue.";
2200 IF PEEK(53279)<>6 THEN 2200
2210 ? CHR$(125):P0$(20)=BALL$:POKE
HP0,50
2220 ? :? "  Here are 4 players and the
missile  group."
2230 ? :? :? "  Notice that the missile group
is a  5th colour."
2240 ? :? " Vertical motion for the missile
group is very easy.":? :? :LIST 2260,2270
2250 DIM B$(9):B$=PM$:B$(2,8)=BALL$:FOR
W=1 TO 500:NEXT W
2260 FOR C=106 TO 16 STEP
-1:MIS$(C)=B$:NEXT C:REM * UP
2270 FOR C=16 TO 106:MIS$(C)=B$:NEXT
C:REM * DOWN
2280 GOSUB WAIT
2300 ? CHR$(125)
2310 ? :? "  END OF PART II"
2320 END
```

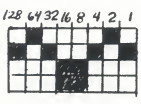
BITMAPS

BALL$




```
32+16+8+4 = 60
64+32+16+8+4+2 = 126
ALL BITS ON  = 255
ALL BITS ON  = 255
ALL BITS ON  = 255
64+32+16+8+4+2 = 126
32+16+8+4 = 60
```

BIRD$




```
64+2 = 66
128+32+4+1 = 165
16+8 = 24
16+8 = 24
```

ACTION
A FIRST LOOK

reviewed by Jerry Aamodt

For those of you who attended our August meeting, ACTION is not unfamiliar. For those who didn't have the opportunity, ACTION is a new language developed by Optimized Systems Software. Bill Wilkenson of OSS was good enough to bring a copy to our meeting, and give us a quick demo. He also left a copy for us to review. This will be the first look at the language, and, in fact, the total package. To give you a complete rundown of the language after only a couple weeks use would be presumptious. We will therefore review the package over the next few months to give a complete analysis.

My background is primarily in BASIC and a smattering of machine language. My first exposure to ACTION was at our August meeting. As my computer was down for maintenance, I found it necessary to acquire my first exposure to ACTION through the manual that accompanies the cartridge. Like ATARI BASIC, it is cartridge based. After two years of booting sensitive protected disks, I found this refreshing. You don't have a plethora of "BOOT ERRORs" to annoy you!

While I waited for my computer to be repaired, I dug into the ACTION manual. Let me note first that unless you have at least a good background in BASIC, the manual will be difficult. The authors state this, and they mean it. The manual can be broken into three parts; the Editor, the Language and Library, and the Compile /Monitor functions.

Let's take the Editor first. One of the strengths of the package is the Editor. The first 30 pages of the manual deals with the this function. Unlike the ATARI screen editor, ACTION's Editor can be used for program development AND as a word processor! In fact, OSS claims you may never go back to your word processor or ATARI editor. As stated before, there is a lot to be said for a word processor that loads directly from a ROM package. The Editor also has some nice features. Cursor speed is at least twice as fast as I have become accustomed to in other editors. It uses the "window on the text" mode. A feature unique to this editor is that you can create a second window, flip back and forth between windows to work on portions of the text. This feature is nice if you are editing a letter and decide you want to rewrite a block or add a paragraph while reviewing the original text. The size of the windows can be adjusted to allow more or less space in each respective window. Another unique feature is the ability to "tag" a location in the text. You can then go to the tag at any time.

With the exception of the two aforementioned unique characteristics, the rest of the Editor functions work much the same as other word processors. You can edit, insert, delete, search, replace, merge, etc. In fact, it is a pretty good word processor, but there are some shortcomings. The single most distressing feature was my inability to insert or change lines of text conveniently. To merge a line, you may have to break the line. This then gives you two lines of text which must be filled up with text to make your copy look good. Otherwise, you will have to adjust each line subsequent until you come to a natural break, such as a paragraph.

Another disturbing feature, though this is a matter of preference, is that each line scrolls horizontally as you enter it. This, while lines above and below remain stationary. The only time this would disturb you is while you are attempting to line up columns. To compensate for this discomfort, you can set tabs, which will allow you to enter column entries. You then need to scroll the screen left and right to verify positioning. As I said, it's a matter of preference. The lack of an 80 column screen means you will have to suffer discomfort in some functions of word processing. The bottom line is that the ACTION package is not meant to be a word processor, but the screen editing function built in to the package provides a very good approximation of one.

The reason you'll buy ACTION will be to acquire a better language for your ATARI.

Having worked for 2 1/2 years with BASIC, I've found that there are several shortcomings to BASIC. The major fault is speed. OSS claims that ACTION will run 1,000 times faster than BASIC. To support the contention that ACTION is fast, they have included the BYTE MAGAZINE "primes" benchmark test program and results as a separate appendix. More about the speed later.

A greater fault of BASIC to a large group out there is that BASIC is not a structured language. To overcome this fault, ACTION has incorporated "the good points of both 'C' and Pascal". What we have here is a structured language, with a speed approaching that of machine language. What has made the language attractive to the BASIC programmer is that a lot of the Library functions can be called up as procedures, which allows you to use your knowledge of BASIC. The ACTION vocabulary supports all of the following:

```
AND     INT     UNTIL   <
ARRAY   LSH     WHILE   <=
BYTE    MOD     XOR     $
CARD    MODULE  +       ^
CHAR    OD      -       @
DEFINE  OR      *       (
DO      POINTER /       )
ELSE    PROC    &       ,
ELSEIF  RETURN  %       [
EXIT    RSH     !       ]
FI      SET     =       "
FOR     STEP    <>      '
FUNC    THEN    #       ;
IF      TO      >
INCLUDE TYPE    >=
```

This in conjunction with the Library calls, gives you a quite substantial and powerful language.

What about the speed of the language? First, OSS has incorporated an option which allows you to turn off the screen display while the computer is working. If you're really "lookin' for speed" then this will do it! In the following example I left the screen display ON. The test is relatively crude but gives a good example of the speed of ACTION. In the example, we will use a simple FOR/NEXT loop to do a repetative calculation. In BASIC, it looks like this:

```
10 FOR I=1 TO 1000
20 K=2*I
30 L=L+K
40 NEXT I
50 PRINT L
```

Running this program will take slightly over 11 seconds. The same program in ACTION looks like this:

```
PROC Timetest()     ;Simple procedure call

CARD I,K            ;you must declare variables

L=[0]               ;initializing L

FOR I=1 TO 1000     ;standard FOR/NEXT loop


DO                  ;the operation

K=2*I               ;
L==+K               ;same as L=L+K

OD                  ;

PrintC(L)           ;print cardinal value

RETURN
```

If we compile and run, the program outputs its result in slightly over one (1) second. This, again, with the screen ON. I also used Library calls, which unfortunately slows down the operation.

Next month, we'll give a bit more. -- -- --

MACE MEMBERS!

WE NEED VOLUNTEERS

TARICON, the worlds first ATARI convention, is coming up fast. The only thing that isn't coming up fast is VOLUNTEERS. Where are you? We know that you're out there. So come on out of the woodwork. Stand-up and be counted. See PAUL WOOD as soon as posible! If you can't find him wandering around the MACE meeting, try calling Marshall at 338-6837

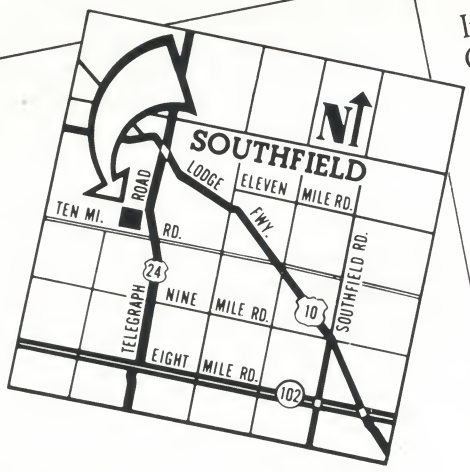
MICRO STATION SOFTWARE CENTER

Come to Micro Station for a huge selection of software books, magazines, printers, monitors, and more . . .

- Data Base
- Word Processors
- Business Packages
- Spread Sheets
- Accounting
- Program Generators
- Graphics
- Entertainment

Sign up now for
ATARI GRAPHICS
and
ATARI BASIC

Taught by M.A.C.E.'s own
Jim Spitzer
Classes start in September



Introductory Printer Special	209.96
Gorilla Banana	26.96
Blue Max	26.96
Shadow World	26.96
Necromancer	30.96
S.A.G.A. #3	30.96
Jumpman Junior	30.96
Type Attack	30.96
Alien Garden	19.96
Preppie	22.96
Castle Wolfenstein	33.96
E.T. Phone Home	54.96
Chess	

OPEN MEETING NIGHTS

Monday-Friday 11-7 Saturday 11-5

Sale prices not combinable with other Specials

24484 W. Ten Mile
(½ blk. W. of Telegraph)
Southfield, MI 48034
(313) 358-5820

micro station

STAR MERCHANTS

Originally from Creative Computing
By Lloyd Johnson

Revised with graphics and sound
By Richard Gizynski

Star Merchants is a trading adventure game with sound and graphics. The scenerio starts with a newly rented space ship heading for Earth with 2,000,000 credits operating capital -- object: buy cargos at a low price that will become more valuable on other planets in the trading area. Then parly your fortunes to become the richest merchant in the galaxy. But beware! Your crew wants a piece of the action and is willing to strike to get it!

Another hazard is one faced by all merchants. Cargos don't always go up in price. Try to buy low (minimum 40%) and sell high (maximum 400%). Remember that you must have enough money to pay your crew and port expenses or the port authorities won't let you off to continue your adventure.

The rules are printed out at the beginning of the game. Next come a list of commands you will need to continue your adventure. Write these commands down on a seperate piece of paper to avoid having to use command 7 -- list available commands.

After landing on earth, command two will let you see what cargos are available and at what price. Command three allows you to buy a cargo. Another command three, another cargo. Be careful that you don't use up all the space on your ship with low profit cargos.

Command five will let you look at the list of starports. Choose wisely based on the type of cargos you have bought. Command six will activate your your rockets for the destination you put in. You're on your way!

When you arrive, command one will let you see the new value of the goods in your hold at the new port. Win some, lose some. Command four will allow you to sell a cargo. The command can be repeated as often as necessary. Be careful, for as your riches grow, the greedy crew may want 'a piece of the action'.

You can negotiate with them. A quick settlement can be had by meeting their demand, but a shrewd trader like you will profit by offering them a compromise between what they were getting and what they were asking for. Don't be too hard for their position will become more resolute as negotiations drag on and they will only settle for an amount closer to their original demand. It all depends on you.

Eventually your lease will run out. You can renew for 2,000,000 or get a summation of your success at game's end. Each adventure you'll try to better your record.

For those of you that are learning to program, most of the routines are easy to follow. Note especially the routine to put commas (lines 3990-4060) in numbers. Also note lines 1030-1040. They act like a simulated 'PRINT USING' function of some other BASICS.

The original Hewlett-Packard BASIC had the trip between stars take about 12 minutes to process. Applesoft would be just about as bad. That is the segment where all the DATA statements were restored and read to get to those chosen at random for new cargoes or those that priced the cargos at a new port. The ability of ATARI BASIC to RESTORE to a line number speeded this to about 30 seconds. The rocket sound and star field covered this time segment and give the impression of a trip.

For those of you who want to tinker, this program leaves a lot of room for add-ons. You might try adding a star pirates segment around lines 2760 (where the interstellar flight ends) or maybe a tax collector on the selling routines that start at line 1530. How about a war bonus for goods run through a blockade on a war-racked planet. Those of you into graphics might like to upgrade the star field and add a doppler effect.

Well, I've done my share for this month. Write to the new editor of MACE and let him know how you made out.

```
10 REM STAR MERCHANT
20 REM BY LLOYD JOHNSON
25 REM ORIGINALLY APPEARED IN
CREATIVE COMPUTING
30 REM REVISED FOR ATARI COMPUTERS
BY
40 REM RICHARD GIZYNSKI
50 GRAPHICS 2+16:COLOR 3:SETCOLOR
4,8,0:SETCOLOR 0,1,12
60 POSITION 8,5:? #6;"STAR":POSITION
6,7:? #6;"MERCHANT"
70 GOTO 4080
80 DIM A$(20),B$(8),C$(13)
90 DIM A(5,4),D(10),H(20,4),P(14),
S(6),T(10)
100 OR=2920:SN=3010:CN=3120:
SD=3490:CD=3600:C=4000
110 D(5)=4.3:B2=2000000:E1=50000:
E2=100000:E3=500000:N4=0:S1=5:
T5=0:T6=0:T7=0:W1=200:W2=200
120 FOR I=1 TO 20:H(I,1)=0:NEXT I
130 RESTORE 3970:FOR I=1 TO 14:READ
X:P(I)=X:NEXT I
140 GRAPHICS 0
150 POSITION 2,8:? "YOU HAVE JUST
SPENT 2,000,000 CREDITS ON A 2 YEAR
LEASE FOR A MERCHANT STAR-SHIP. ";
160 ? "THIS LEAVES YOU WITH 2 MILLION
CREDITS OPERATING CAPITAL.":?
170 ? "YOUR SHIP CAN HOLD A TOTAL OF
20":? "CARGOS WITH A TOTAL";
180 ? " CARGO WEIGHT OF  200 TONS.
THE FUEL CAPACITY OF YOUR"
190 ? "SHIP IS GREAT ENOUGH SO THAT
TRAVEL  BETWEEN ANY";
200 ? " TWO STARPORTS IS POSSIBLE
WITHOUT REFUELING."
210 POSITION 10,22:? "PRESS RETURN TO
CONTINUE ";:INPUT A$:? "?"
220 ? :? :? "YOU ARE PRESENTLY
TRAVELING FROM ALPHACENTAURI TO
SOL.";
230 ? " YOU ARE CARRYING NO CARGO."
240 ? :? "THE STARPORT TRADE
CLASIFCATION DETER-MINES THE
CARGO";
250 ? " PRICE BUT DOES NOT":?
"DETERMINE WHICH CARGOS ARE
AVAILABLE."
260 ? :? "ABBREVIATIONS USED FOR
TRADE CLASS":? "ARE AS FOLLOWS:":?
270 ? "     R=RICH":? "     P=POOR":
? "     I=INDUSTRIAL":
? "     NI=NONINDUSTRIAL"
280 ? "     A=AGRICULTURAL":
? "     NA=NONAGRICULTURAL"
290 ? :? :? "        PRESS RETURN TO
CONTINUE";
300 INPUT A$
310 ? "?":? :? :? :? "COMMANDS
AVAILABLE ARE AS FOLLOWS:"
320 GOSUB OR
330 ? :? "PRESS RETURN TO
CONTINUE":INPUT A$:GOSUB 1880
340 REM ** MAIN LOGIC **
350 IF STRIKE=1 THEN 380
360 IF RND(1)>0.01+3.0E-03*B2/E3 THEN
380
370 GOSUB 680:?
380 ? "ACCOUNT BALANCE ";:
B2=INT(B2):A$=STR$(ABS(B2)):
IF B2<0 THEN ? "-";
390 GOSUB C:? A$:? "EMPTY CARGO SPACE
";W2
400 ? "SHIP TIME ";T6;" YEARS."
410 ? "ENTER COMMAND";:
TRAP 410:INPUT C1:?
420 IF C1>0 AND C1<7 AND
C1=INT(ABS(C1)) THEN 450
430 GOSUB OR
440 GOTO 410
450 ON C1 GOSUB 930,1070,
1220,1530,1680,1770
460 B3=B2+T1
470 IF B3<0 THEN 570
480 IF T6-2*T5<2 THEN 340
490 ? :? "THE LEASE HAS EXPIRED ON
YOUR SHIP"
500 IF B3<2000000 THEN 570
510 ? "ANOTHER TWO YEAR LEASE WILL
COST":? "2,000,000 CREDITS"
520 ? "DO YOU WISH TO RENEW YOUR
LEASE(Y/N)";
530 INPUT A$:IF A$="N" THEN 570
540 LET B2=B2-2000000
550 T5=T5+1
560 TRAP 410:GOTO 340
570 REM END PROGRAM
580 G1=INT((B3-4000000)/T6)
590 B$="LOSS"
600 IF G1<0 THEN 620
610 B$="GAIN"
620 IF B3>0 THEN 640
630 ? :? "YOU NO LONGER HAVE
SUFFICIENT FUNDS TOOPERATE YOUR
SHIP"
640 ? :? "YOU STARTED WITH 4,000,000
CREDITS"
650 ? "YOU NOW HAVE ";:
A$=STR$(ABS(B2)):GOSUB C:IF B2<0
THEN ? "-";:? A$
```

```
660 ? "CARGO IN HOLD IS WORTH ";:
A$=STR$(T1):GOSUB C:? A$
670 ? :? "THIS REPRESENTS A ";B$;"
OF":A$=STR$(ABS(G1)):GOSUB C:? A$;:? "
CREDITS PER YEAR":GOTO 4070
680 REM ** STRIKE SUBROUTINE **
690 GRAPHICS 2+16:COLOR 1:SETCOLOR
4,4,0:SETCOLOR 0,8,9:SETCOLOR 2,4,0
700 POSITION 7,6:? #6;"STRIKE"
710 FOR X=1 TO 100:SOUND
3,4+X/4,2,7-X/20:SOUND
0,10+X/2,2,5-X/20:SOUND
1,3+X/4,2,5-X/20:SOUND 2,20+X/2,2,4
720 NEXT X
730 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND
2,0,0,0:SOUND 3,0,0,0
740 FOR X=1 TO 300:NEXT X:GRAPHICS 0
750 GRAPHICS 0:? "THE CREW HAS GONE
ON STRIKE"
760 ? "YOU ARE CURRENTLY PAYING
THEM ":? E3;" CREDITS ANNUALLY"
770 E5=INT(E3+0.5*RND(1)*E3)
780 ? "THEY ARE ASKING FOR ";:
A$=STR$(E5):GOSUB C:? A$;" CREDITS
ANNUAL SALARY"
790 N4=0
800 E6=E3+N4*(E5-E3)/10
810 ? :? "ENTER COUNTER OFFER";
820 TRAP 810
830 INPUT E7
840 IF E7>=E5 THEN 900
850 IF E7<E6 THEN 870
860 IF (E5-E7)/(E5-E6)<RND(1) THEN 900
870 ? "OFFER REJECTED - TRY AGAIN"
880 N4=N4+1
890 GOTO 810
900 ? "OFFER ACCEPTED":STRIKE=1
910 E3=E7:TRAP 410
920 RETURN
930 REM ** LIST HOLD **
940 ? "CARGO STORED IN HOLD":POKE
85,34:? "% OF"
950 IF W2<W1 THEN 980
960 ? "ALL PARTITIONS ARE EMPTY"
970 RETURN
980 ? "HOLD DESCRIPTION  AMT  PRICE
BASE"
990 FOR I=1 TO 20
1000 IF H(I,1)=0 THEN 1050
1010 RESTORE CN+10*H(I,1)
1020 READ A$
1030 POKE 85,2:? I;:POKE 85,5:?
A$;:A$=STR$(H(I,3)):POKE 85,24-LEN(A$):?
A$;:A$=STR$(H(I,2)):GOSUB C
1040 POKE 85,35-LEN(A$):?
A$;:A$=STR$(H(I,4)):POKE 85,39-LEN(A$):?
A$
1050 NEXT I
1060 RETURN
1070 REM ** LIST CARGO FOR SALE **
1080 ? "CARGOS AVAILABLE FOR
PURCHASE"
1090 IF N1>0 THEN 1120
1100 ? "ALL AVAILABLE CARGOS HAVE
BEEN":? "PURCHSED"
1110 RETURN
1120 ? "LOT                    % OF"
1130 ? "NO  DESCRIPTION   AMT
  PRICE  BASE"
1140 FOR I=1 TO 5
1150 IF A(I,1)=0 THEN 1200
1160 RESTORE CN+10*A(I,1)
1170 READ A$
1180 POKE 85,2:? I;:POKE 85,5:?
A$;:A$=STR$(A(I,3)):POKE 85,24-LEN(A$):
? A$;:A$=STR$(A(I,2)):GOSUB C
1190 POKE 85,35-LEN(A$):?
A$;:A$=STR$(A(I,4)):
POKE 85,39-LEN(A$):? A$
1200 NEXT I
1210 RETURN
1220 REM ** BUY CARGO SUB **
1230 ? "ENTER THE LOT NUMBER OF
CARGO YOU WISH";"TO PURCHASE";
1240 INPUT K
1250 IF K=ABS(INT(K)) AND K>0 AND K<6
THEN 1280
1260 ? "INVALID LOT NUMBER"
1270 RETURN
1280 IF A(K,1)>0 THEN 1310
1290 ? "LOT ";K;" HAS BEEN ALREADY
PURCHASED"
1300 RETURN
1310 IF A(K,2)<=B2 THEN 1340
1320 ? "YOU CANNOT BUY CARGO ON
CREDIT"
1330 RETURN
1340 IF A(K,3)<=W2 THEN 1370
1350 ? "YOU DO NOT HAVE SUFFICIENT
CARGO SPACE"
1360 RETURN
1370 FOR I=1 TO 20
1380 IF H(I,1)=0 THEN 1420
1390 NEXT I
1400 ? "ALL 20 CARGO PARTITIONS ARE
OCCUPIED"
1410 RETURN
1420 FOR J=1 TO 4
1430 LET H(I,J)=A(K,J)
1440 NEXT J
1450 A(K,1)=0
1460 W2=W2-H(I,3)
```

```
1470 B2=B2-H(I,2)
1480 T6=T6+3.0E-03
1490 N1=N1-1
1500 ? "TRANSACTION COMPLETED"
1510 ? "CARGO STORED IN PARTITION ";I
1520 RETURN
1530 REM ** SELL CARGO SUB **
1540 ? "ENTER PARTITION OF CARGO TO
BE SOLD";
1550 INPUT K
1560 IF K=ABS(INT(K)) AND K>0 AND K<21
THEN 1590
1570 ? "INVALID PARITION NUMBER"
1580 RETURN
1590 IF H(K,1)>0 THEN 1620
1600 ? "CARGO PARTITION EMPTY"
1610 RETURN
1620 LET B2=B2+H(K,2)
1630 W2=W2+H(K,3)
1640 T6=T6+3.0E-03
1650 H(K,1)=0
1660 ? "TRANSACTION COMPLETED"
1670 RETURN
1680 REM ** LIST STARPORTS SUB **
1690 ? "STAR          TRADE":? "NO NAME
CLASS";
1700 ? " DISTANCE DIR."
1710 RESTORE
1720 FOR I=1 TO 10
1730 READ A$,B$
1740 ? I;:POKE 85,5:? A$;:POKE 85,21:?
B$;:POKE 85,29:? D(I);:POKE 85,36:? T(I)
1750 NEXT I
1760 RETURN
1770 REM ** TRAVEL SUB **
1780 IF B2>0 THEN 1810
1790 ? "YOU CANNOT LEAVE STARPORT":?
"UNTIL ALL DEBTS ARE CLEARED"
1800 RETURN
1810 ? "ENTER DESTINATION STAR
NUMBER";:INPUT I:IF I<>S1 THEN 1840
1820 ? "YOU ARE ALREADY AT ";
I;:RESTORE SN+10*I:READ A$:? " ";A$
1830 RETURN
1840 IF I=ABS(INT(I)) AND I>0 AND I<11
THEN 1870
1850 ? "INVALID STAR NUMBER"
1860 RETURN
1870 LET S1=I
1880 GRAPHICS 5+16:SETCOLOR
4,0,0:SETCOLOR 0,8,9
1890 PLOT 9,3:PLOT 37,3:PLOT 22,4:PLOT
15,7:PLOT 69,8:PLOT 40,9:PLOT 68,10
1900 PLOT 14,11:PLOT 15,13:PLOT
73,14:PLOT 66,15:PLOT 75,15:PLOT
4,17:PLOT 55,18:PLOT 35,19
1910 PLOT 15,22:PLOT 49,22:PLOT
71,23:PLOT 22,31:PLOT 53,31:PLOT
33,35:PLOT 7,36
1920 PLOT 6,37:PLOT 53,37:PLOT
71,38:PLOT 19,39:PLOT 27,41:PLOT
45,42:PLOT 17,43:PLOT 3,45
1930 FOR X=1 TO 100:SOUND
0,10+X,8,9-X/25:NEXT X
1940 REM *** GET STAR TRADE &
LOCATION
DATA **
1950 RESTORE SD+10*S1
1960 READ X3,Y3,M1
1970 FOR I=1 TO 6
1980 M2=2^(6-I)
1990 LET S(I)=INT(M1/M2)
2000 LET M1=M1-S(I)*M2
2010 NEXT I
2020 REM ** APPRAISE CARGO IN HOLD **
2030 LET T1=0
2040 FOR I=1 TO 20
2050 IF H(I,1)=0 THEN 2220
2060 RESTORE CD+10*H(I,1)
2070 LET D2=0
2080 FOR J=1 TO 6
2090 READ D3
2100 LET D2=D2+D3*S(J)
2110 NEXT J
2120 READ B1
2130 LET P2=INT(12*RND(1)+D2):IF PLAY<5
AND P2<4 THEN P2=P2+3
2140 IF PLAY<2 THEN P2=P2+2
2150 IF P2>1 THEN 2170
2160 P2=1
2170 IF P2<14 THEN 2190
2180 P2=14
2190 H(I,4)=P(P2)*100
2200 H(I,2)=P(P2)*B1*H(I,3)
2210 T1=T1+H(I,2)
2220 NEXT I
2230 REM ** GET CARGOS FOR SALE **
2240 LET N1=5
2250 FOR I=1 TO 5
2260 D2=0
2270 LET T2=INT(36*RND(1))+1
2280 A(I,1)=T2
2290 RESTORE CD+T2*10
2300 FOR J=1 TO 6
2310 READ D3
2320 LET D2=D2+D3*S(J)
2330 NEXT J
2340 P2=INT(12*RND(1))+D2:IF PLAY<5
AND P2>10 THEN P2=P2-4
2350 IF PLAY<2 THEN P2=P2-1
2360 IF P2>1 THEN 2380
2370 P2=1
```

```
2380 IF P2<14 THEN 2400
2390 P2=14
2400 A(I,4)=P(P2)*100
2410 READ B1,Q1
2420 Q2=1
2430 IF T2<17 THEN 2470
2440 Q2=5
2450 IF T2<32 THEN 2470
2460 LET Q2=10
2470 LET Q3=0
2480 FOR J=1 TO Q1
2490 Q3=INT(6*RND(1))*Q2+Q3
2500 NEXT J
2510 A(I,3)=Q3
2520 A(I,2)=B1*Q3*P(P2)
2530 NEXT I
2540 REM ** DISTANCE AND DIRECTION
OF STARS
2550 D4=D(S1)
2560 RESTORE SD
2570 FOR I=1 TO 10
2580 READ X1,Y1,D9
2590 X2=X1-X3
2600 Y2=Y1-Y3
2610 IF X2<>0 THEN 2670
2620 IF Y2<0 THEN 2650
2630 LET T(I)=90
2640 GOTO 2740
2650 LET T(I)=270
2660 GOTO 2740
2670 T(I)=ATN(Y2/X2)*180/3.14159:
T(I)=INT(T(I))
2680 IF X2>0 THEN 2700
2690 LET T(I)=T(I)+180
2700 IF T(I)<360 THEN 2720
2710 LET T(I)=T(I)-360
2720 IF T(I)>0 THEN 2740
2730 T(I)=T(I)+360
2740 D(I)=SQR(X2^2+Y2^2):
D(I)=INT(D(I)*10+0.5)/10
2750 NEXT I
2760 REM ** ARRIVAL EXPENSES AND
STAR NAME **
2770 T6=INT(100*(T6+0.02*D4+0.02))/100
2780 E4=(T6-T7)*E3:PLAY=PLAY+1
2790 T7=T6
2800 FOR X=100 TO 1 STEP -1:SOUND
0,10+X,8,9-X/25:NEXT X
2810 FOR X=1 TO 50:NEXT X:FOR X=1 TO
9:SOUND 0,10,8,9-X:NEXT X
2820 RESTORE SN+10*S1
2830 READ A$,B$
2840 ?
2850 ? "YOU HAVE ARRIVED AT ";A$
2860 ? "EXPENSES HAVE BEEN DEDUCTED
AS FOLLOWS"
2870 ? "DOCKING FEE ";:
A$=STR$(E1):GOSUB C:? A$
2880 ? "FUEL ";:A$=STR$(INT(E2*D4)):
GOSUB C:? A$
2890 ? "CREW SALARY ";:A$=STR$(E4):
GOSUB C:? A$:FOR X=1 TO 100:NEXT X
2900 B2=B2-(E1+E2*D4+E4):STRIKE=0
2910 RETURN
2920 ? "COMMAND","DESCRIPTION"
2930 ? "  1","LIST CARGO IN HOLD"
2940 ? "  2","LIST CARGO WHICH":? ,"MAY
BE PURCHASED"
2950 ? "  3","BUY CARGO"
2960 ? "  4","SELL CARGO"
2970 ? "  5","LIST STARPORTS"
2980 ? "  6","TRAVEL TO NEW STAR"
2990 ? "  7","LIST AVAILABLE
COMMANDS"
3000 RETURN
3010 REM ** STAR NAME DATA **
3020 DATA LALANDE 21185,NI-P
3030 DATA ALPHA CENTAURI,NA-I
3040 DATA SIRUS,A
3050 DATA BARNARDS STAR,I-P
3060 DATA SOL,R
3070 DATA ROSS 154,NI-NA
3080 DATA EPSILON ERIDANI,A-P
3090 DATA LUYTEN 726-8,NA
3100 DATA LUYTEN 789-6,A-NI-P
3110 DATA ROSS 248,A-I
3120 REM ** CARGO NAMES **
3130 DATA CRYSTALS
3140 DATA RADIOACTIVES
3150 DATA SPECIAL ALLOYS
3160 DATA PHARMACEUTICALS
3170 DATA GEMS
3180 DATA AIRCRAFT
3190 DATA GRAV SLEDS
3200 DATA COMPUTERS
3210 DATA ATV
3220 DATA AFV
3230 DATA FIREARMS
3240 DATA AMMUNITION
3250 DATA PLASMA GUNS
3260 DATA TOOLS
3270 DATA BODY ARMOR
3280 DATA FARM MACHINERY
3290 DATA LIQUOR
3300 DATA SILVER
3310 DATA SPICES
3320 DATA ELECTRONIC PARTS
3330 DATA MECHANICAL PARTS
3340 DATA CYBERNETIC PARTS
3350 DATA COMPUTER PARTS
3360 DATA MACHINE TOOLS
3370 DATA SPACE SUITS
```

```
3380 DATA FRUIT
3390 DATA TEXTILES
3400 DATA POLYMERS
3410 DATA MEAT
3420 DATA PETROCHEMICALS
3430 DATA GRAIN
3440 DATA WOOD
3450 DATA COPPER
3460 DATA TIN
3470 DATA STEEL
3480 DATA ALUMINUM
3490 REM ** STAR DATA **
3500 DATA 2.83,-7.36,20
3510 DATA -2.4,-3.56,9
3520 DATA 8.38,9.93,2
3530 DATA -6.1,0,24
3540 DATA 0,0,32
3550 DATA -8.87,2.05,5
3560 DATA 8.45,6.65,18
3570 DATA 2.99,7.42,1
3580 DATA -4.43,9.3,22
3590 DATA -.89,10.26,10
3600 REM ** CARGO DATA **
3610 DATA 3,-2,2,-2,0,-4,20000,1
3620 DATA 0,1,4,-3,0,-2,1000000,1
3630 DATA -2,0,-4,6,1,-2,200000,1
3640 DATA -1,4,-4,3,-2,0,100000,1
3650 DATA 4,-2,4,-4,-1,1,1000000,1
3660 DATA -2,4,-3,3,1,-1,1000000,1
3670 DATA 2,0,-1,1,0,0,6000000,1
3680 DATA 1,0,-2,0,0,0,10000000,1
3690 DATA -2,2,-2,1,1,0,300000,1
3700 DATA 0,2,-2,0,0,1,700000,1
3710 DATA -2,6,-4,1,0,0,30000,2
3720 DATA -1,6,-5,2,0,0,30000,2
3730 DATA -1,3,-2,0,0,0,200000,2
3740 DATA -4,7,-8,4,5,0,10000,2
3750 DATA -3,6,-4,1,0,0,50000,2
3760 DATA -2,2,-6,0,6,-4,150000,1
3770 DATA 3,3,-1,0,-3,0,10000,1
3780 DATA 3,-1,3,-1,0,-2,70000,1
3790 DATA 4,-2,3,-1,-5,2,6000,1
3800 DATA 0,0,-4,4,1,1,100000,1
3810 DATA 0,1,-3,3,2,1,75000,1
3820 DATA 1,0,-4,2,1,0,250000,1
3830 DATA -1,0,-2,3,0,0,150000,1
3840 DATA 1,0,-2,1,0,0,750000,1
3850 DATA -1,2,-3,2,2,0,400000,1
3860 DATA 1,2,3,3,-4,-6,1000,2
3870 DATA 3,0,-3,1,-5,-3,3000,3
3880 DATA -2,0,3,3,0,0,7000,4
3890 DATA 0,0,5,2,-5,5,1500,4
3900 DATA 2,0,4,-2,3,0,10000,6
3910 DATA 0,0,1,3,-5,6,300,8
3920 DATA 0,0,1,2,-7,3,1000,2
3930 DATA -1,0,-2,3,0,0,150000,1
3940 DATA 1,0,-2,1,0,0,750000,1
3950 DATA -1,2,6,0,0,0,500,4
3960 DATA -1,1,3,-2,0,-2,1000,5
3970 REM ** PRICE DATA **
3980 DATA .4,.5,.7,.8,.9,1,1.1,
1.2,1.3,1.5,1.7,2,3,4
3990 REM ** COMMAS IN NUMBERS **
4000 L=LEN(A$):IF L<4 THEN RETURN
4010 Z=1:Z1=1:IF L/3=INT(L/3) THEN 4040
4020 IF L/3-INT(L/3)>0.5 THEN
C$=A$(1,2):C$(3)=",":Z=3:Z1=4:GOTO 4040
4030 C$=A$(1,1):C$(2)=",":Z=2:Z1=3
4040 C$(Z1)=A$(Z,Z+2)
4050 IF Z+2=L THEN A$=C$:RETURN
4060 C$(Z1+3)=",":Z1=Z1+4:Z=Z+3:GOTO 4040
4070 TRAP 50000:END
4080 REM ** OPENING MUSIC **
4090 QT=4180:HLF=4190:WHL=4200
4100 FOR A=180 TO 243 STEP 6:SOUND
0,A,10,8:NEXT A:FOR X=1 TO 5:NEXT X:
SOUND 0,243,10,8:GOSUB HLF
4110 SOUND 0,162,10,8:GOSUB WHL:SOUND
0,182,10,8:GOSUB QT:SOUND
0,193,10,8:GOSUB QT:SOUND
0,217,10,8:GOSUB QT
4120 SOUND 0,121,10,8:GOSUB HLF:SOUND
0,162,10,8:GOSUB WHL
4130 SOUND 0,182,10,8:GOSUB QT:SOUND
0,193,10,8:GOSUB QT:SOUND
0,217,10,8:GOSUB QT:SOUND
0,121,10,8:GOSUB HLF
4140 SOUND 0,162,10,8:GOSUB HLF
4150 SOUND 0,0,0,0:SOUND
0,182,10,8:GOSUB QT:SOUND
0,193,10,8:GOSUB QT
4160 SOUND 0,182,10,8:GOSUB QT:FOR S=8
TO 1 STEP -1:SOUND 0,217,10,S:FOR X=1
TO 5:NEXT X:NEXT S:SOUND 0,0,0,0
4170 GOTO 80
4180 FOR X=1 TO 10:NEXT X:SOUND
0,0,0,0:RETURN
4190 FOR X=1 TO 20:NEXT X:SOUND
0,0,0,0:RETURN
4200 FOR X=1 TO 40:NEXT X:SOUND
0,0,0,0:RETURN
```

MODIFY THE NEW 810 DISK DRIVE TO USE BOTH SIDES OF A DISK

by Christopher Hansen

Are you tired of punching holes into your Disks so that you can write to the back side? Well then, read on!

This article deals only with the New 810 Disk Drives. But if you own an older model, read it anyway. The ideas presented here just might help you modify your older Drive.

The new model was released about the first of the year and you can tell it from the older model by the fact that you have to manually pull up the front to insert a disk. Whereas the older models have a button that you push to open the door. Thats the difference on the outside, there is quite a difference on the inside as you might expect.

Before I go to much further, I had better give you a few WORDS OF WARNING. You will void the warranty on your 810 Disk Drive if you open it up to do the modifications that follow. This includes both the regular 90-day warranty and any extended service contracts that you may have paid for. Also, the Author assumes no responsibility for damages resulting from attempting the modifications outlined in this article.

There are a lot of different ways to do what I'm about to explain. But most of them deal with drilling a hole or holes into the case of the Drive. Since I just might want to restore my Drive to its original condition, I decided to modify it so it could be put back the way it was and no one would know the difference. That is unless they took off the cover and looked inside.

First of all, some basics. When you open the door of the Drive and insert a disk, there is a lever that moves up into the slot of the disk (remember, I'm talking about the new Drives, not the old ones that used a beam of light). This lever is actually part of a switch. When the lever is in the up position, the contacts of the switch are closed, thus allowing you to write to the disk. When the lever is in the down position, the contacts are open, and there is no way you can write to the disk. This happens when there is a write protect tab over the slot in the disk, or there is no slot present at all. What we are going to do is by-pass the lever switch altogether and install our own switch.

For those of you that are not electrically inclined, or those of you who just don't want to bother with the wiring of the new switch, I will also describe a much simpler modification. The modification is an easy and relatively safe operation as long as you take your time and be VERY CAREFUL.

Be sure to clean your work area thoroughly to avoid contaminating the Drive mechanism. Also, you'll want to check the tools you'll be using to make sure that they are not magnetized. If a tool won't pick up a staple, then it's safe to use it.

If you're all set, lets get at it. First of all locate the four circular screw-hole covers on top of the Drive and carefully pry them off with a penknife or razor. Once you get them off, you can stick them to the top edge of the cover so that you won't lose them. Next use a Phillips head screwdriver to remove the four screws that hold the cover on. There are no internal clips holding the cover in place, so you can now gently lift the cover off of the base of the Drive and set it aside. Be sure not to lose the four screws.

You are now looking at why the 810 Disk Drive costs so much. At least that's Atari's story. Avoid touching any of the components or dropping anything inside of the Drive. You will find the lever switch on the left towards the front of the Drive. Locate the two white wires leading from it and trace them to the back of the Drive. They will terminate into a plug that plugs into a circuit board on the left rear side of the Drive.

If you ever plan on restoring your Drive to its original condition, now is the time to label the two wires. Label the wires about an inch and a half back from the plug.

YOUR ONE STOP MARKETPLACE FOR ALL YOUR COMPUTER NEEDS.

RWE RITE WAY

COMPUTERS • CASH REGISTERS • SOFTWARE

ATARI

star micronics · inc

MAKING A NAME FOR OURSELVES

PRINTERS

OWN YOUR OWN PHONE!

COMPUTER SOFTWARE

HAYES *Modems*

300 Baud	$239
1200 Baud	$549
Micro Modem II	$299

GEMINI-10X $299

GEMINI 15.....$399

COLUMBIA
DATA PRODUCTS, INC.

CompuServe

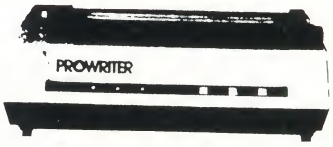
SOFTWARE & HARDWARE

OVER **3000** PROGRAMS STOCKED

AUTHORIZED SERVICE FOR
- Atari • Franklin
- Gemini • Prowriter
- R.C. Allen • Epson

EPSON

Pro/Writer



$399.

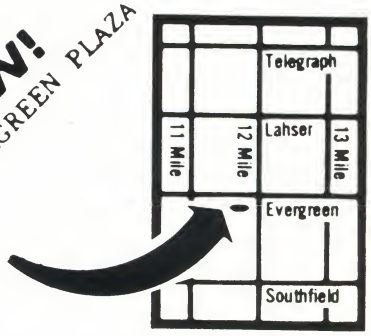
We carry a full line of Cables and Accessories

HIGH PERFORMANCE FLOPPY DISK $1.59

LOWEST PRICES

OPEN MONDAY THRU FRIDAY 10-9, SATURDAY 10-6

NOW! IN THE EVERGREEN PLAZA



RITE WAY ENTERPRISES INC.

COMPUTERS • SOFTWARE • CASH REGISTERS
SALES • SERVICE • SUPPLIES

(313) 751-2454
8262 12 MILE RD.
WARREN, MI 48093

(313) 559-2980
19783 12 MILE RD.
SOUTHFIELD, MI 48076
IN THE EVERGREEN PLAZA

(313) 562-3178
22027 MICHIGAN AVE.
DEARBORN, MI 48124

What do we say about competition? We welcome it!!!

You'll want to put two labels on each wire far enough apart to allow you to cut and strip the wires. I used 'TOP' and 'BOTTOM' to label the wires in my Drive.

Now the fun starts. NOTE the position of the plug in the circuit board and VERY GENTLY pull it out. When you have the plug out, take a pair of wire cutters and cut the two wires between the labels. Don't strip the wires leading from the lever switch. Just wrap some friction tape around each wire and then around both wires together. Make sure that they won't drop down onto the circuit boards in the bottom of the Drive.

Now that that's done, we can turn our attention to the two wires leading from the plug. Strip these wires back about one half of an inch. For those of you that really didn't want to get involved into wiring in a new switch, all you have to do now is twist these two wires together. You can solder them together while you're at it, if you feel it is necessary. I personally think it is wise to do so. Whether or not you do solder them, at least wrap some friction tape around the bare wires. Now remembering how you took the plug out to begin with, GENTLY push it back onto the circuit board the same way. Thats it. Just put the cover back on the way you took it off and you will be able to write to both sides of ANY disk. BUT REMEMBER, from now on you will have to lock all of the files that you want to protect from being written to or deleted.

For those of you that are going to install your own switch, I'm going to assume that you know what you're doing. For those of you that aren't all that sure of yourself, it might be best to get a friend, or someone who does know what he's doing, to help you. It's really not all that hard, so bear with me.

You'll need the following items to finish the modification. They can all be bought at Radio Shack. Of course you don't have to get the exact items. These are just the ones I used on my Drive. They are:
1) Cat.# 278-1303 Stranded 18 gage hook-up wire. $2.19
2) Cat.# 61-2745 6' Extension cord. $1.39
3) Cat.# 275-666 DPDT Standard toggle switch. $1.29
4) Cat. #272-703 Neon panel lamp assemblies, pkg. of 2. $2.19
5) Cat.# 270-233 4"x2"x1 5/8" Experimenter box. $1.69
6) Cat.# 278-1638 Spiral wrap. $2.19
7) Cat.# 64-001 Rosin core electronic solder, .5-oz. Handy pak. $.89
8) Cat.# 270-026 Quick-disconnect plug. $.99
9) Cat.# 64-3025 Vinyl Grommets. $.99

That comes to a grand total of $13.81 plus tax. Not too bad of a price to pay to have a professional looking job when you are through. But like I said, you don't have to use the items that I used. You might even have some of them at home already.

O.K., back to the job at hand. Do not solder the two plug wires together as in the method above. Instead, figure out how far away you will have your control box from the Drive, and cut two lenghts of 18 gage wire that long. My box is right next to my Drive, so my lead wires were only about 16" long. Solder these two lead wires to the wires on the plug. Feed the two lead wires through the hole in the back of the Drive where you adjust the Drive number by moving the little white and black levers. Don't worry, there's still enough room to move the levers.

Take the bottom off of the experimenter box to make it easier to drill the holes for the DPDT switch and neon lamps. Drill a 7/16" hole in the center of the top of the box for the DPDT switch. Drill two 5/16" holes in one side of the box for the two neon lamps. I drilled mine on center line about 1 1/4" in from each end of the box. That way you'll miss the switch in the center of the box. On the opposite side of the box, as close to the bottom as possible, drill at least a 3/8" hole for the wiring to go through. Take a file and file the hole through to the bottom. That way you won't have to feed the wires through it. Deburr all the holes with a penknife or file, or whatever you have that will make the plastic box smooth. Set it aside for now.

Solder one of the lead wires from the Drive plug to a center lug on the DPDT switch. Solder the other lead wire to one of the end lugs on the same side. That side of the switch is now complete.

Solder about an 18" or so lead wire to one lug of each of the neon lamps. Now solder a 6" lead wire to the other lug on each of the lamps. Set these aside for now.

On the opposite side of the DPDT switch that you soldered the Drive plug wires to, solder about an 18" lead wire to the center lug. Now mount the DPDT switch in the hole on top of the box. Make sure the lever of the switch moves toward the front and back of the box, not end to end. Tighten it down, but remember that it is only a plastic box. Try to keep the wires out of your way by keeping them in the hole in the back of the box.

Feed the lead wires from the neon lamps through their respective holes, and solder one 6" lead wire from each lamp to an end lug on the DPDT switch. Make sure that they are on the opposite side of the Drive plug side of the switch. Push the lamps all the way into the holes and tighten them down.

If you have gotten this far without throwing the box across the room, congratulations. IT IS A VERY TIGHT FIT. We're just about home free.

Now cut the female end of the extension cord off. Separate the wire and strip the two ends back about 1/2". Solder the two long lead wires from the neon lamps to one of the extension cord wires, and solder the long lead wire from the DPDT switch to the other.

Make sure that all the wires leaving the box are going through the hole in the back of the box. You can now screw the bottom plate back onto the box.

Now back to the Drive plug. Thought I had forgotten it, didn't you? Remember how it came off? Well, GENTLY push it back onto the circuit board the same way. Find a place inside of the Drive away from everything and secure the two plug lead wires to it with some tape or a tie-down. Put the cover of the Drive back on the way you took it off, and tighten it down. Put the little covers back over the screw holes.

All thats left now is to wire in the quick-disconnect plug. If you ever move your Drive you don't want the control box hanging around. I installed mine right at the back of the Drive. Wherever you decide to do it, just cut the two wires and solder it in line.

Now wrap the spiral wrap around the wires coming from the Drive, and the wires going to the extension cord. Plug everything in, and you now have your own write protect switch. With the switch turned on, you can write to both sides of a disk. And when the switch is turned off, you can't write to any side of the disk. I like that last part, because my two young sons are always playing games on the Atari. And before I had my switch, I was always afraid that they might accidentally erase something of valve.

There is only one thing left to do. Thats to glue four little vinyl grommets over the screws on the bottom of the control box to keep the box from sliding around. And of course, you'll have to make note of what color neon lamp is on when you are writing to a disk.

I hope these directions haven't been to confusing. I tried to make them as plain as possible. Happy hacking.

-- -- --

GETTING NOISY

Complaints are being heard around town concerning the noise level at MACE meetings. It's great to socialize and share information with friends you haven't seen for awhile, but how about making it available to all of us by becoming part of the program. All of us came to hear the latest and if you've got something good ...

-- -- --

YOUR AD

Could have run in this spot. See! We had to use it as a filler. It's a free service of MACE to members. Just write an ad to sell or tell and send it to your friendly editor.

BEST OF BAKER STREET

IN THE BEGINNING

By RICHARD GIZYNSKI
July, 1982

Learning to program is a lot like learning how to type. Try to learn by small, overlapping steps. Baker Street is going to explore some beginning and intermediate questions new programmers have. Just as you can learn to drive a car without knowing its engineering details, you can learn to use your computer without knowing its details. But knowledge can turn your computer from 'sedan' to 'sports model'.

To those of you that are new to Atari programming, there are two excellent books and a short manual that will help you get started. The soft cover 'ATARI BASIC -- A Self-Teaching Guide' is a book that every beginner should own. It takes you through a step by small step exploration of fundamental BASIC commands. It also provides short, easy to understand examples of these commands.

The second book you should consider is Stimulating Simulations by C.W. Engel. This is a collection of game programs for the beginner to type in. The collection demonstrates, again in step by step fashion, how a program is organized, written and expanded. There aren't any graphics routines in the book, but entry level programmers shouldn't concentrate on graphics.

The third selection is a super time saver called 'Master Memory Map' by Santa Cruz Educational Software. Memory Map is a listing of the most useful memory locations in your Atari and a very short description of what they do. It is organized in numerical sequence and is a handy quick referance.

Now let's explore some terms that may be new or foggy to you. Two terms that pop up when we talk about memory are "bits" and "bytes". A bit is the smallest unit of information that a computer can handle. A byte is a grouping of eight bits. I'll show you the how and why of these.

Computer memories consist of thousands of small areas that can be turned on or off. These areas may be miniature circuts, magnetic 'bubbles' or blips or holes punched in paper. All that is necessary is that they can be created and read individually or in organized groups. They are the 'bits' of a computers memory. They can be (and are) represented by a one or a zero. A one usually means that a bit is on or present. A zero means a bit is off or absent.

To form more complex units of information, eight bits are strung together to form a byte. By using eight bits, we can represent numbers up to 255. Starting at a bit we call the least significant bit (the one on the furthest on the right like our normal decimal number system) we consider each bit as an exponent of two. The right most bit has a value of one (or a place marker like the zero in 30). The next bit has a value of two. The third bit has a value of four. Each additional bit is two times the previous bit (or a place marker). The most significant bit (the left-most bit) of a byte has a value of 128. By using this system, we can represent a number from zero (all place markers) to 255 (all bits on). Count 'em -- 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1.

Why and how bytes are used is easier to understand if you think of our alphabet and numbering system. We have 10 digits that can be strung together to form a number of any size. Our alphabet has 26 capital and 26 lower case letters that do the same thing with words. Both our alphabet and our numbering system also use special characters. Periods, commas, dollar signs and a large number of other symbols are added to the basic digits and alphabet characters to form our written language.

You can use a number to represent each of the characters, digits, symbols and the spaces that seperate them into meaningful groups. When you look at your Atari keyboard, you see 56 keys a space bar and four special purpose buttons. Add to this all of the characters that are available by using two or three key combinations (shift and control characters). You need 255 seperate numbers to represent all of the

possible combinations. You also need a zero to mark a 'no information' condition.

One more hurdle to go and understanding your computer gets a little easier. When we look at a number between 0 and 255, we see one or two or three digits. Not so for the computer. It 'sees' that number the same way we see an alphabet character. When we look closely at a character, we can see how it is formed. A's and H's are formed using three lines but arranged differently. Different bytes are formed using eight bits but arranged differently. A computer "sees" the eight bits of a byte at the same time, just as we see the parts of an alphabet character at the same time. The computer has a 255 (plus zero) character alphabet.

We can think of the words that make up this article as a code to understanding an idea. In a similar manner, the computer takes each byte as a unit of instruction or data. Since each byte can hold up to 256 different messages, instructions can be handled by using a particular byte-value for each instruction. A sort of super short hand. With 255 different instructions, your computer can do fantastic things.

Data, such as numbers or words and symbols, use the same set of byte-characters that are used for instructions. Computer instruction and data must be arranged in the correct sequence to be understood by the machine. First, an instruction to tell the computer where to get information. Next comes the information to be processed and then another instruction. All the instructions that the computer can handle are in byte code.

-- -- --

TELL THE WORLD

Discovered something new about your Atari? Had fun with a new game. Angry about a product or a dealer? Want to share your programming success with us.

This magazine is your forum for discussion. The editors are always hungry for copy. If it reasonable, we'll probably print it.

ASSEMBLER LANGUAGE SIG

By Todd Meitzner, Secretary, SIGASM

AUGUST MEETING

This months meeting was attended by a smaller number of people than last month but included several people new to the sig group. We all introduced ourselves and went on with the meeting.

The talk this month was by Phil Heavin and was about a movement calculating routine. It was very interesting and Phil even included a source code listing and a disk of the routines he used.

After that Phil showed how he used INSTEDIT to edit some of the charactors he used in one of his demos. Some discussion was also done on a problem one of the members had in a program.

The discussion for the September meeting (already taken place as you read this) is to be about debugging a program.

OCTOBER'S MEETING

The meeting place for October's meeting is to be at George Van Houw's house in Sterling hts. It will take place as usual on the first Thursday, October 6th. It will start at 7:00 with socializing with the actual business portion begining at 7:30. You may contact George at 268-3331 or me at 542-1752 for further info. Hope you are there.

EDUCATIONAL SIG

Our next meeting will be held Sept 28 at 7:30 at Warren Tower High School. Plans for TARICON will be finalized and a workshop on P.M. graphics will be conducted. Please call Mark Davies (774-9709) to reserve a place in the workshop.

ACROSS THE POND

by Mike Lechkun

This article will introduce a new and hopefully regular department to the MACE JOURNAL. We hope to use this column to further ideas and experiences with other ATARI users and groups and the ATARI community worldwide. We would like to encourage anyone to write down your thoughts about how you use your machine, or about computing here in the U.S. We will forward your article to other newsletters not on this continent to be printed in their newsletters. Here's your chance to be internationally published!

Send articles to MACE c/o Newsletter Editor, PO BOX 2785, Southfield, MI 48037.

This month's article comes to us from "PAGE 6 - An ATARI Magazine", published by ABACUS (the Around Britain Atari Computer Users Society). This is an excellent publication which truly rivals our own MACE JOURNAL in quality. If you get a chance, examine our copies at this or the next meeting (see me).

Disk Directory - by Colin Friston

One of the most common uses for DOS on your disks is to obtain a directory of the contents. How many of us can remember all the file names we make up? DOS on every disk will take up 81 sectors, but here is a program that will give you a directory on screen or on the printer at a cost of only 11 sectors per disk.

I usually file as "DIR" so that it can be accessed simply by RUN"D:DIR". The program opens an IOCB to the disk drive directory and runs between lines 40 and 60 getting all directory entries. At the end of file the program traps to line 90 and gives an optional printout. A trap to line 100 is included to cover the printer being out of commission. The program is a fast way to recall and record what is on each disk without having to go into DOS.

```
1 REM ,*****************************,
2 REM *      DISK DIRECTORY       *
3 REM *          by               *
4 REM *      COLIN FRISTON        *
5 REM *                           *
6 REM *      Reprinted from       *
7 REM *  "PAGE 6 -An ATARI(tm)    *
8 REM * magazine"  Feb/Mar '83, #2 *
9 REM '*****************************'
15 DIM A$(20),B$(20):TRAP 70:OPEN #1,
6,0,"D:*.*"
20 GRAPHICS 0:POKE 752,1:POKE 712,176
:POKE 710,176:? "      DISK NUMBER D
IRECTORY"
21 REM 7 SPACES BEFORE WORD 'DISK'
22 REM ELIMINATE THESE 2 REM'S
3         0                  ?                    "
MMMMMMMMMMMMMMMMMMMMM":R
EM 07 SPACES,21 CTRL M'S FOR UNDERLIN
E
40 INPUT #1;A$
50 P=P+1:IF P=21 THEN POKE 82,20:POSI
TION 20,3
60 ? A$:GOTO 40
70 CLOSE #1:POKE 82,2:OPEN #2,4,0,"K:
"
80 POSITION0,23:? "         HIT Y FO
R PRINTOUT      ";:GET #2,K:IF K<>89
THEN 170
81 REM LINE 80 INVERSE VIDEO BETWEEN
QUOTES
90 TRAP 100:LPRINT :GOTO 110
100 POSITION 7,23:? "PRINTER DOES NOT
 RESPOND !";:FOR T=1 TO 1000:NEXT T:G
OTO 80
110 TRAP 160:OPEN #1,6,0,"D:*.*"
120 LPRINT :LPRINT "         **
 NUMBER    **"
130 LPRINT "         ** DISK DIRECT
ORY **":LPRINT
140 A$=" ":B$=" ":INPUT #1;A$,B$:LPRI
NT A$;"   ";B$
150 GOTO 140
160 LPRINT A$;"   ";B$
170 CLOSE #1:CLOSE #2:GRAPHICS 0
```

-- -- --

USE 'LPRINT' BEFORE CSAVE

Why you ask. Because you clear out anything that happens to be in the buffer and are less likely to lose your program. Don't worry about the noise, its because you haven't got a printer.

BEST OF BAKER STREET

PRE-BASIC

By RICHARD GIZYNSKI
December, 1982

In my last article, I described how memory was held in individually read areas or addresses. Each address holds eight bits that form a byte (a character of information). There are 256 possible combinations that can be held by each byte. Now lets see how these characters can be used.

The heart of your Atari is a little silicon chip called the 6502. Another name for this chip is CPU or Central Processing Unit. The CPU, like a handyman for hire, looks for an instruction and then follows it. When it is done with the first instruction it looks for the next.

Each time you turn your Atari on, the CPU first goes to the permanent, Read Only Memory (ROM) section of memory called the Operating System. The operating system contains the instructions on how to handle the screen, get information from the keyboard, check to see what devices are attached, and all the housekeeping chores that the CPU has to perform to allow you to interact with your Atari.

Each instruction must be given to the CPU in a very simple, one step at a time form, like you might give a blind man to guide him through a maze. The CPU has no way of sensing the difference between an instruction byte and a data byte so the first location in memory must be an instruction. That instruction must tell the CPU what to do first and then be followed by another instrution. Or, the first instruction must tell the CPU where to go in memory to get the next instruction. The programer must be careful not to direct the CPU to a byte of data (a number, letter of a word etc.) when the CPU should be looking for an instruction. The CPU would take the data byte and treat it as an instruction and faithfully try to execute it. This level of instructions, the level the CPU works at, is called machine language.

Writing machine language instructions is a very tedious and frustrating task. Programers developed a language that was more human oriented, called assembler, to help them. Assembler languages act like a interpreter. A programmer can write in characters and numbers and the assembler translates this to the binary bytes that the CPU must have. To a beginning programer, assembler is very confusing but much better than looking at columns of 0's and 1's.

BASIC is one of the high level languages. In Assembler, you write each detailed step the CPU needs but use letters and characters that are easier for you to understand. In higher level languages you may use a one or two word command. The language translates this command into the many small steps the CPU needs to act on the command.

In your Atari, the Basic Cartridge contains a program. It is the program in the Basic cartridge that is running. The program that you entered through the keyboard, cassette or disk is really data that the cartridge program will use to do what you want done. The fact that the program that you enter is data for the program in a cartridge is one of the most misunderstood (or completely unknown) concepts to a beginning programer.

When you enter a Basic program, the program in the cartridge stores the program you enter in shorthand form. The form is called tokenization. One token, one command. When you LIST a program to screen, printer, cassette or disk, the program in the cartridge looks at the series of tokens and re-translates them back to English. This is an important concept for you to understand. It will make understanding Basic and your Atari easier. The program that you enter is data for the program cartridge.

VISICALC COMMAND GUIDE

To initiate a command type a / followed by a single letter to denote main menu selection. This is usually followed by a letter from the sub menu and occasionally a third letter for a sub-sub-menu selection will be necessary.

The following abbreviated command reference chart has the main menu selection letter in bold face followed by a brief description of its meaning in light face. Indented under it and in light face are sub choices.

B Blanks out memory cell

C Clear entire worksheet (requires Y to confirm)

D Delete entries
- R Deletes all enteries in row
- C Deletes all entries in column

F Format (appearance of worksheet item)
- D Defaults to Global format of current window format
- G General — the style the sheet starts with
- I Interger — decimal figures are truncated
- L Left justified or flush left
- R Right justified or flush right
- $ Dollar and cents style — with two decimal places
- * Displays stars in bar graphs

G Global — commands that follow affect all or large portions of the sheet
- C Column width can be set between 3 and screen width
- O Order of calculation by:
  - R Rows
  - C Columns
- R Recalculation of figures controlled either:
  - A Automatically
  - M Manually
- F Format change which follows will take effect over entire sheet (see individual format commands under F)

I Insert a blank row or column for good visual effect
- R Row
- C Column

M Move — enter starting co-ordinates.ending co-ordinates

P Print — upper left co-ordinate at cursor, enter lower-right co-ordinate

R Replicate or duplicate — press return key to pick up single cell to be duplicated. If you wish to duplicate a block of cells, press period followed by ending co-ordinate for a block replication,- Followed by a colon then the beginning co-ordinate period ending co-ordinate of the destination for the duplicates.

S Storage — disk control commands
- L Load file (followed by file name)
- S Save file (followed by file name)
- D Delete file on the disk (Y to verify command)
- I Initialize or format disk
- Q Quit — end the program followed by slot number of disk controller
- # Save or load partial worksheet
  - S Save — Needs a file name followed by lower right hand co-ordinate of area to be saved followed by a R or C to determine if save will be by Row or Column order
  - L Load — Same as save but inputs from previously saved file

T Title control — keeps titles in place as you scroll screen
- H Horizontal — fixes all titles above cursor
- V Vertical — fixes all titles to left of cursor
- B Both — fixes both vertical and horizontal titles
- N Negates or cancels fixed positioning of titles

V Version number of VisiCalc program

W Window control — allows screen to be split so you can look at widely seperated areas at the same time
- H Horizontal — Horizontal split at the cursor
- V Vertical — Vertical split at the cursor
- 1 One window expanded to full screen
- S Start syncronized scroll capability
- U Unsyncronized scroll

- Fills column with character that follows (to improve appearance)

Michigan Atari Computer Enthusiast Constitution

Adopted September 15, 1981

I Purpose of organization

It shall be the purpose of the Michigan Atari Computer Enthusiasts (hereafter called MACE, or the Club) to promote an interchange of ideas and information concerning the Atari personal computers. Mace shall be organized as a non-profit Michigan corporation.

II Membership and Dues

There shall be no requirements for membership in MACE other than an active interest in Atari computers.

The club, by majority vote of the general membership, may levy upon the entire membership such dues or assessments as shall be deemed necessary for the conduct of the club's business. Every member must pay these assessments when due, in order to retain all of the rights and privileges of membership in MACE.

III Organization and Procedure

Sec. 1 The Executive Board (Board or EB) shall be the chief administrative body of MACE.

A. Voting members shall be:
1. President
2. Vice President
3. Secretary
4. Treasurer
5. Publications Director
6. Program Director
7. Software Librarian(s)

B. Members ex officio shall be:
1. The chairperson of any committee
2. Members of the previous year's Board

C. Each voting member of the Board shall have one vote, the Executive Board shall determine its own rules of procedure, and elect its own chairperson.

Sec. 2 The Executive board shall organize, plan and administer club activities. Thje board shall carry out the business of the club between general membership meetings, its decisions being subject to tacit approval of the general membership.

Sec. 3 The President and/or Treasurer may authorize expenditures of club money up to $50 per month. The Executive Board as a whole may authorize expenditures of up to $100 per month. Any expenditures of club funds above $100 in any month must be authorized by the majority vote of the general membership.

Sec. 4 The Club, by majority vote at any general membership meeting, may establish provisional or standing committees to expedite the mangement of club activities. Each committee shall determine its own rules of procedure and membership, govern its own affairs and activities, and elect a chairperson and other officials as deemed necessary.

Sec. 5 All proceedings not covered in this constitution or bylaws shall be goverened by the latest edition of Roberts Rules of Order. The Executive Board shall determine all questions of constitutionality arising within the club, and interpret all unclear clauses of this constitution, bylaws, and all club resolutions.

IV Election of Officers

Sec. 1 The elected offices of MACE shall be:
1. President
2. Vice-President
3. Secretary
4. Treasurer
5. Publications Director
6. Program Director
7. Librarian(s)

Sec. 2. Officers shall be elected for a term of one year by ballot of the members present at the general membership for the month of September, provided there is a quorrum. For purposes of these elections, a quorum will be constituted by no less the 40% of the general membership.

Committee elections will be held at the last committee meeting held in the month of September.

Sec. 3 Nominations shall be open to all members. In all elections, winners shall be determined by simple majority. If there is no winner on the first ballot, the name of the candidate with the least number of votes shall be withdrawn, and a second ballot taken. This process shall be repeated until one candidate achieves a simple majority of the vote.

Vacancies occuring between elections must be filled by special election to be held at the first meeting following the creation of the vacancy.

V Duties of Officers

Sec. 1 The President shall preside at all general membership meetings, and conduct them according to the rules adopted. The president shall enforce observance of the Constitution and Bylaws, and perform the customary duties of the office, as stated in the latest edition of Roberts rules of Order.

Sec. 2 The Vice President shall assume the duties of President in the absence of the latter, and shall be an ex-officio member of all committes and coordinate inter-committee activities. In addition, the Vice-President shall function as Advertising Manager of the Newsletter.

Sec. 3 The Secretary shall keep a record of the proceedings of the club. He/She shall function as chief communications officer, and shall be responsible for the gathering and dissemination of information of interest to the general membership. He/She shall carry on all club correspondence, and be responsible for the preparation of documents on the club's behalf.

Sec. 4 The Treasurer shall carry out the club's financial funds, and keep an accurate account of each transaction, and of the club's current financial status, as well as assisting in the process of corporate financial reporting.

Sec. 5 The Publications director, shall be responsible for the production of the monthly newsletter, and any other club publication. (The Vice-President shall act as advertising manager for the newsletter)

Sec. 6 The Program Director shall have the responsibility for seeing to it that a suitable meeting place is available for the general membership meetings, and for planning and organizing the program of those meetings.

Sec. 7 The Software Librarian(s) shall be responsible DONEFOR MAINTAINING THE Software libraries and related documentation, as well as copying and distributing library disks and tapes.

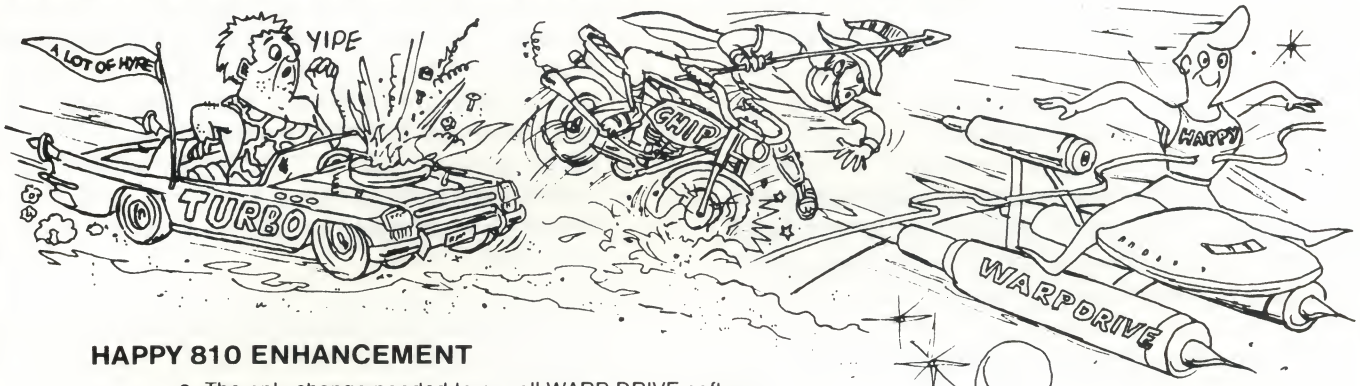
VI Amendment of Constitution

This constitution may be amended by a vote of 2/3 of the members at a general membership meeting at which there is a quorum of 51% of the membership in attendance.

VII Unmentioned Powers

All powers, privileges, rights, and duties not otherwise delegated by this Constitution shall be reserved for the general membership of this club.

HAPPY WINS THE RACE WITH WARP DRIVE SPEED!



HAPPY 810 ENHANCEMENT

- The only change needed to run all WARP DRIVE software
- Plug in P.C. board requires no permanent modifications
- Proven reliable in thousands of installations, reduces disk drive wear
- Comes completely assembled and tested, just plug in and use
- Full one year parts and labor guarantee, compatible with existing software
- High quality printed circuit board with gold connectors

NO ONE ELSE HAS THIS PERFORMANCE
Unenhanced whole disk (ATARI rev B format) read time: 112 seconds
Unenhanced whole disk (ATARI rev C fast format) read time: 89 seconds
ENHANCED 810 whole disk (any format) read time with standard software: 68 seconds
ENHANCED 810 whole disk (any format) read time with WARP DRIVE software: 43 seconds
Standard software whole disk write and verify time: 238 seconds
WARP DRIVE software whole disk write and verify time: 62 seconds

NEW HAPPY WARP DRIVE SOFTWARE

WARP SPEED HAPPY BACKUP PROGRAM
- Completely automatic: nothing to figure out, insert disks and press return
- Only program on the market guaranteed to backup any disk
- Can write to a blank disk: format write and verify in one operation
- Automatic program tracing: copies only the tracks that are used
- Efficient memory utilization: reduces the number of disk insertions
- Requires only one ENHANCED disk drive, backups will work on a standard drive

WARP SPEED MULTI DRIVE HAPPY BACKUP PROGRAM
- Same features as above plus support of multiple ENHANCED drives
- Can be used with up to 4 ENHANCED drives
- Source and all destination drives read and write in parallel
- Format write and verify 3 complete disks in less than 3 minutes

WARP SPEED HAPPY COMPACTOR PROGRAM
- Reduces the number of disks required to backup your library
- Combines up to 8 self booting disks into 1 disk with a menu
- Compacted disks run only on an ENHANCED drive
- Pays for itself by saving on disks
- Single or dual ENHANCED drive operation

HAPPY WARP DRIVE DOS
- Improves ATARI DOS 2.0S to use warp speed reading and write with verify
- Use all features of BASIC, PILOT, FMS, and DUP at top warp speed
- Warp speed I/O software module available separate from DOS

HAPPY WARP DRIVE SECTOR COPY PROGRAM
- Standard format whole disk read, write and verify in 105 seconds
- Use with sngle or dual drives, mix ENHANCED and NON-ENHANCED drives

HAPPY CUSTOMIZER PROGRAM (sold separately $99.95)
- Creates custom format disks of any specification
- Any type bad sector, duplicate sector numbers, or interleave
- Easy to use but requires an advanced level user to interpret the results

Authorized Distributor:
KIRK'S MICRO SALES
P.O. BOX 2145
LIVONIA, MICH. 48151
(313) 933-7030
24 Hour MODEM Line
(313) 427-1402

REVIEWED IN POPULAR MAGAZINES

A.N.A.L.O.G. COMPUTING—July/August 1983 *"...The installation instructions for the Happy 810 Enhancement are among the best I have ever seen. ...The Happy 810 Enhancement is one of the most powerful hardware modifications available to ATARI computer owners."*

ANTIC—July 1983 *"The difference between a normal ATARI 810 disk drive and one equipped with Happy is like the contrast between mass transit and the automobile. A car costs you more initially, but improves the quality of your life. Similarly, if you use your disk drive a lot, installing Happy will markedly enhance your programming life."*

SPECIAL SUGGESTED RETAIL PRICE BEFORE DECEMBER 31, 1983: *Get the HAPPY 810 ENHANCEMENT with the single and multi drive HAPPY BACKUP PROGRAM, plus the HAPPY COMPACTOR PROGRAM, plus the HAPPY DRIVE DOS, plus the HAPPY SECTOR COPY, all with WARP DRIVE speed, including our diagnostic for $249.95. Existing registered ENHANCEMENT owners may upgrade to WARP DRIVE speed for $15.00 with no hardware changes.!*

Price includes shipping by air mail to U.S.A. and Canada. Foreign orders add $10.00 and send an international money order payable through a U.S.A. bank. California orders add $16.25 state sales tax. Cashiers check or money order for immediate shipment from stock. Personal checks require 2-3 weeks to clear. Cash COD available by phone order and charges will be added. No credit card orders accepted. ENHANCEMENTS for other ATARI compatible drives coming soon, call for information. Please specify -H model for all drives purchased new after February 1982, call for help in ENHANCEMENT model selection. Dealers now throughout the world, call for the number of the dealer closest to you.

HAPPY COMPUTING • P.O. Box 32331 • San Jose, California 95152 • (408) 251-6603

ARMOR ASSAULT

Review by
ALAN SCHOEN

I was asked to write something short for the MACE magazine. I told the "Editor" that I don't know how to write -- but he won't listen, so here it goes ready or not.

I recently bought a game called "Armor Assault" that was put out by EPYX. I thought that it was going to be an interesting game, since it had different scenarios, the ability to play solitary or two player, the ability to plot mines, the ability to plot opportunity or direct fire, the ability to modify maps and units.

When you start playing the game, you will probably try a solitary game. Don't be surprised if you do badly. But if you can get the russians to line up or get into a group, they don't know enough to go around each other and not to try to fire through each other. If you can destroy one Russian tank and get the destroyed tank between you and the other Russians, they will become weak by trying to continue their attack by going through the destroyed tank, which no other tank can go through.

In the different scenarios, if there are any mountains or rubble and if you can put one of your tanks onto it, then the only way the enemy can shoot you is with direct missile fire, whereas you can use opportunity fire if they are not on anything.

When you decide to modify a map, you will be referring to the manual, but to your surprise the book tells you how to set the walls of buildings but not the inner part of a building (of which might be one of many errors of the game). If you fool around with the game as much as I have, you will find out that a CTRL "M" will set the inside part of a building. So then you start setting a building and to your surprise when you want to set the top wall of a building, the book says CTRL, but CTRL what? Again by fooling around I discovered that CTRL "," will set the top wall.

When you want to set into a map some water, you will find out that the book is somewhat helpful, but again not complete. Where the book says SHIFT "S" is not what you want but SHIFT "$" is. Then again the book says SHIFT "=" is really SHIFT "3", the book says SHIFT ";" is really SHIFT "5". Other than the printing errors in the book, it can be enjoyable to create your own scenario.

If you want to set any rubble you use SHIFT ",".

If you play the game often enough, you might get bored with it by playing solitary, but it might be more fun by playing two player, but I don't yet know.

-- -- --

NOTICE

BERKLEY COMPUTER FAIR.

Saturday, Oct. 8th 11 AM TO 5 PM. BERKLEY HIGH SCHOOL, 2325 Catalpa (11 1/2 Mi.) at Coolidge. Hardware and software displays and demonstrations. see home computers as applied to education, business, home management and entertainment. Free admission -- open to the public -- refreshments.

We also need creative, self written programs, to illustrate the comprehensive applications of home computers. If you're interested in demonstrating or helping as a volunteer, call MIKE ELSEY at 547-2034 (AFTER 6 PM). Sponsored by the Berkley Parent Advisory Council on Instructional Computing.

-- -- --

HAVE YOU SEEN

The new expanded science section in every Thursday Detroit News. This section was first proposed by one of our members. It has several good articles and is a sign of the growing awareness of the role micro's in the Detroit area.

BEST OF BAKER STREET

BEGINING A PROGRAM

By RICHARD GIZYNSKI
February, 1983

Many beginning programers have their first programing experience trying out their machines. They start by copying short routines from an instruction manual. Next comes experiments in putting a few BASIC commands together on their own. Frustration grows as new programmers run out of uses for the commands they have learned. If this sounds familiar, this article is written for you.

If you suddenly moved to a country with a different language, you would learn words with the most immediate meanings first. Computer equivalents might be: PRINT - "Talk to me"; CLOAD - "Listen to these instructions"; RUN - "Do it to it". As you expand your BASIC vocabulary, keep in mind that, like any stranger to a new language, you don't have to learn all the uses of a word to start communicating.

You can save yourself a lot of frustration by taking a few minutes to write, in English, what you want the computer to do. This helps clarify your thoughts and helps you to tackle larger programs. Programing a computer is like giving instructions to an obedient idiot. The computer will try to do what you tell it, in the order you tell it to follow. If you can't tell yourself what you want done, you won't be able to tell the computer.

I start a program by writing something like "An easy way to keep my checkbook balanced" or "A way to keep phone numbers," as starting ideas. Next, I add details to the ideas. "Add new names and numbers," "retrieve stored names and numbers," and "alphabetize by name" are examples of expanding a program description. After noteing all the features I want, I weed out the rediculous one and just keep those that are easy to nearly impossible.

Complicated tasks are made a lot easier by breaking them into small pieces. After I have a working description of what I want the program to do, I begin to organize the steps I need to get the job done. This is called flow charting. In ordinary English terms, I make block diagrams of what steps are needed. For Telephone Directory, "Imput name", "Input phone number" and "add new name and number to the a list of names in memory" might form my first block diagrams. This makes it easy to see what I am going to ask the computer to do.

At this point, I start looking at more detailed features needed in the program. Under "Input new info" I might put "Check to see if input was a name or a code telling the program to go to another routine." "Make sure that the name and number are flush left and right on the screen" might be added to "Printing routine".

So far I haven't written any program code but I do have a clearer picture of what I am going to ask the computer to do. I can spot groups of repeated steps that might be put into subroutines and "bugs" in my logic BEFORE I actually try to write the program.

After I finish the flow chart I start writing the program itself. Following the flow chart plan, I give variables names that relate to their function. PLAYERNAME$ is easier to trace through a program than P$. REMark statements head up each area of the new program. Six months from now I want to be able to look at the program and quickly find out what I was doing.

With a detailed flow chart, writing the code is almost easy. If you don't know how to do a particular operation, you at least know what to look or ask for. The Atari Basic manual has examples of how many command words can be used. You can find other command examples in programs written here in MACE or in magazines and books available at your nearest computer store or at your library.

The most important point in beginning a program is to describe what you want the program to do. Start with the main purpose then add features to it. Next, start a rough outline of how the computer will do the job. Then break down each rough outline step into smaller, more detailed steps. As you clarify your thoughts you will find the program getting much easier to understand and write.

```
**************************************
*          OTHER ATARI BBSs          *
*        FROM A.R.C.A.D.E. BB        *
*            313-978-8087            *
*          UPDATED 07/24/83          *
*  NOTES:                            *
*  L = LIMITED Evenings & Weekends   *
*  R = RING-BACK Call, Call-back     *
*  # = ORIGINAL of This format       *
*  & = 1200 Baud optional            *
*  * = 24 HOUR Operation             *
*      Listed by Location            *
*                                    *
*Please send Updates/Corrections to:*
*   A.R.C.A.D.E. BBS 313-978-8087    *
* Specify: as per the below format!  *
```

ST	AC-NUMBER	TYPE	NAME	NOTE
AL	205-979-0513	AMIS	B.A.C.E.	L
CA	213-271-0224	AMIS	TIME ZONE	?
CA	213-783-8373	AMIS	L.A.A.C.E.	*
CA	408-253-5216	AMIS	GFX	*
CA	408-298-6930	AMIS	IBBS	L
CA	408-732-1079	?	VANVISION	*
CA	415-527-8276	ARMU	SYNAPSE	L
CA	619-447-8143	AMIS	S.D.A.C.E.	L
CA	714-731-6523	?	A.C.A.O.C.	?
CA	714-973-2086	ARMU	ORANGE CTY.	*
CA	805-498-2508	AMIS	THOUSAND OAKS	?
CA	805-528-0957	?	SLO COUNTY	*
CA	805-544-8173	?	SLOPOKES	*
CA	916-363-3304	AMIS	A.C.C.E.S.S.	*
CA	916-487-3679	AMIS	S.A.C.S BBS	L
CO	303-221-1779	TARI	POOR RICHARD	L
CO	303-758-2927	AMIS	DENVER	?
CO	303-758-6233	AMIS	DENVER	L
DC	202-276-8342	ARMU	WASHINGTON	*#
DC	202-364-8617	AMIS	ALADDINS LAMP	?&
GA	404-252-9438	ATAB	ROD R.	#
IA	515-961-8881	AMIS	MID AMERICA	*
IL	312-620-6441	AMIS	S.C.A.T.	*
IL	312-789-0499	RBBS	CHICAGO	*
IL	312-789-3610	AMIS	CHICAGO	*
IL	312-889-1240	AMIS	C.L.A.U.G.	*
IL	312-893-6884	?	FoReM	*
IL	312-925-2929	?	WIZ-BANG S.	*
IN	219-665-2650	AMIS	LAND O LAKES	*
IN	219-665-6225	AMIS	ANGOLA BBS	L
LA	504-273-3116	RBBS	BATON ROUGE	*
LI	516-666-2148	?		L
MA	617-266-7789	BULT	BOSTON BULLET	*
MA	617-595-0211	AMIS	NORTH SHORE	*&
MA	617-667-7388	AMIS	MACRO EXCH.	*
MD	301-974-4987	TARI	SEVERNA PARK	*
MD	301-587-2132	ARMU	COMPUTER AGE	L
MD	301-871-5370	?	RICKY MOOSE	?
MD	301-972-4444	AMIS	WARLORDS	?
MI	313-274-3940	AMIS	M.A.C.E. W.	*
MI	313-352-8271	?	SY-DRAFT	*
MI	313-427-1402	AMIS	CAPTAIN KIRK	L
MI	313-538-0197	AMIS	DARTBOARD	L
MI	313-544-0885	AMIS	M.A.C.E.	*#
MI	313-547-7903	?	CPU AT & AP	?
MI	313-759-6569	CBBS	ROYAL OAK	*R
MI	313-771-4126	AMIS	St Cl SHORES	L
MI	313-978-8087	AMIS	A.R.C.A.D.E.	*
MI	517-371-4321	AMIS	C.H.A.O.S.	*
MI	616-241-1971	AMIS	G.R.A.S.S.	*
MO	314-432-7034	AMIS	ComputerCave	L
MO	314-535-3799	?	AURA BBS	?
MO	314-928-0598	?	TERMINAL END	*
NC	919-692-7710	AMIS	A.C.C.E.S.S.	*
NJ	201-339-7407	AMIS	BAYONNE	L
NJ	609-267-7825	TARI	BOB ALLEGER	L#
NJ	609-924-5875	TARI	ELECT. CANDY	L
NV	702-733-9488	AMIS	LAS VEGAS	L
NY	212-241-8965	AMIS	SPIDER WEB	*
NY	212-927-6919	AMIS	DANTE'S INF.	*
NY	212-767-6633	AMIS	INFO 800	*
NY	212-798-2027	ARMU	WEEKENDS	L
NY	212-860-4940	AMIS	LATE NIGHT	L
OH	216-271-9273	AMIS	GARFIELD	L
OH	216-439-5543	AMIS	NORTH COST	L
OH	419-423-0206	AMIS	FLAGCITY	*
OH	614-476-2035	?	ATRPOST DEMO	*
OK	405-636-0218	AMIS	SOFT.SAFARIS	*
OK	405-681-6929	AMIS	TEBBTOWI	?
OK	405-685-2027	AMIS	SW. SAFARIS	L
OK	405-722-5056	ARMU	GREKELCOM	?
OK	918-836-4742	AMIS	SAT.EVE ONLY	L
OR	503-343-4352	ARMU	A.C.E.	?
PA	215-333-3753	AMIS	J R'S BBS	*
PA	215-398-3937	RBBS	ALLENTOWN	*
PA	215-432-9848	AMIS	STARCADE	*
PA	215-836-5116	RBBS	CHELTENHAM	L&
PA	412-655-2652	ARMU	PACE	*
RI	401-521-1998	AMIS	?	?
TN	615-833-2902	AMIS	ROCKABILLY	L
TX	512-837-2003	AMIS	ARMADILLO	*
TX	817-485-1892	AMIS	Fort Worth	L
TX	817-532-2981	AMIS	CENTEX BBS	L
TX	817-534-6649	?	ATARI ZONE	L
TX	817-589-1254	ARMU	ACUGD	*
TX	817-799-9096	AMIS	WACO TX	L
WA	206-525-3412	AMIS	SPACE	L
WA	206-574-5438	AMIS	ACTION BBS	L
WA	206-632-9473	AMIS	M.B.B.S.	*
WA	509-582-5217	AMIS	A2-D2	L
WA	509-624-2636	AMIS	SPOKANE	L
WI	414-352-2772	AMIS	MIL-ATARI	*
WI	608-251-8538	AMIS	MAGIC LANTERN	*

MICHIGAN ATARI COMPUTER ENTHUSIASTS

P.O. Box 2785
Southfield, MI 48037

24HR BULLETIN BOARDS 544-0885/274-3940

MACE HOTLINE (Meeting Info) 338-6837

PRESIDENT
Marshal Dubin
2639 Hempstead
Auburn Heights, MI 48057
338-6837

VICE-PRESIDENT
Jerry Aamodt
4148 Huhn
Rochester, MI 48063
574-1020

TREASURER
James Phillips
40008 Cambridge
Bldg 23 - Apt 103
Canton Township, MI 48187
981-1523

CORRESPONDING SECRETARY
Mike Lechkun
32229 Ruehle
Warren, MI 48093
978-8432

RECORDING SECRETARY
William Black
1251 Duckwood
Milford, MI 48042
887-6870

PROGRAM COORDINATOR
Gretchen Levitan
12709 Borgman
Huntington Woods, MI 48070
399-6964

DISK LIBRARIAN
Chet Gonterman
35088 Savannah Lane
Farmington Hills, MI 48018
553-7443

CASSETTE LIBRARIAN
Faris Ajo
6642 Leytonstone
West Bloomfield, MI 48033
661-0388

NEWSLETTER EDITOR
Arlan R. Levitan
12709 Borgman
Huntington Woods, MI 48070
Compuserve: 70675,463
Source ID: TCT987
Voice: 399-6964

NEXT MEETING 10/18/83

Southfield Pavillion
Ten & a Half & Evergreen
7:00 PM

M.A.C.E.
P.O Box 2785
Southfield, MI 48037

$20.00 fee for 12 months

M.A.C.E. MEMBERSHIP APPLICATION

Name _____ Phone _____
Street _____
City _____ State _____ Zip Code _____
Company (if applicable) _____
System Description _____ Disk/Tape _____
Suggestions _____
I can help with . . . _____

[ ] NEW [ ] RENEWAL

If Renewal, MACE #_____

*Make checks payable to: M.A.C.E.

Date _____ Coupon _____
Amount _____ Membership Card Number _____
☐ Cash Expires _____
☐ Check Number

Micro Chip is introducing the new gerneration of diskettes from ULTRA MAGNETICS. For maximum reliability each of Ultra Magnetics' disks are 100% surface tested. The Ultra high burnished surface with a special lubrication gives you better head to media contact and longer head and disk life. The disks have a lifetime guarantee and for a limited time are specially priced!

Ultra Magnetics Mini Floppy disks

$18.95

Micro Chip INC.

HOME COMPUTERS AND SOFTWARE

3632 Rochester Rd. • Century Shopping Center • 524-1230

MICHIGAN ATARI COMPUTER ENTHUSIASTS
P.O. BOX 2785
SOUTHFIELD, MICHIGAN 48037

BULK RATE
U.S. POSTAGE
PAID
PERMIT #431
SOUTHFIELD, MI

IMPORTANT DATED MATERIAL

PLEASE DO NOT DELAY

Printing and Bindery Services by • GRAPHIC ENTERPRISES, INC. Detroit, Michigan • 313-839-6800